

IMPROVED INTRA BLOCK WALK AROUND REFRESH FOR H.264

By: John Sievers

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The present invention relates generally to video communication, and more particularly to providing an efficient method of updating a digitally transmitted video image while making efficient use of a given bit budget.

2. Description of Related Art

[0002] Digitization of video images has become increasingly important. In addition to their use in global communication (e.g., videoconferencing), digitization of video images for digital video recording has also become increasingly common. In each of these applications, video and accompanying audio information is transmitted across telecommunication links including telephone lines, ISDN, DSL, and radio frequencies, or stored on various media devices such as DVDs and SVCDs.

[0003] Presently, efficient transmission and reception, as well as efficient storage of video data may require encoding and compression of video and accompanying audio data. Video compression coding is a method of encoding digital video data such that less memory is required to store the video data and a required transmission bandwidth is reduced. Certain compression/decompression (CODEC) schemes are frequently used to compress video frames to reduce required transmission bit rates. Thus, CODEC hardware and software allow digital video data to be compressed into a more compact binary format than required by the original (i.e., uncompressed) digital video format.

[0004] Several approaches and standards to encoding and compressing source video signals exist. Some standards are designed for a particular application, such as ITU-T Recommendations H.261, H.263, and H.264, which are used extensively in video conferencing applications. Additionally, standards promulgated by the Motion Picture Experts' Group (MPEG-2, MPEG-4) have found widespread application in consumer electronics and other applications. Each of these standards is incorporated by reference in its entirety.

[0005] A digital image (501, FIGS. 5A & 5B) is comprised of a grid of individual pixels. Typically, the whole image is not processed at one time, but is divided into blocks that are individually processed. Each block comprises a rectangular grid of a predetermined number of luminance or luma pixels (which generally specify the brightness of a pixel) and a predetermined number of chrominance or chroma pixels (which generally specify the color of a pixel). A predetermined number of blocks are combined into a macroblock (502, FIGS 5A & 5B), which forms the basic unit of processing in, for example, the H.264 standard. Additionally, in the H.264 standard, a group of macroblocks may be combined into a larger processing unit known as a slice (503, FIGS 5A & 5B). Although some aspects of this hierarchy of processing units are discussed below, methods and techniques for block-based processing of images for processing are generally known to those skilled in the art, and thus are not repeated here in detail.

[0006] The blocks of image data may be encoded in a variation of one of two basic techniques. For example, "Intra" coding may be used, in which the original block is encoded without reference to historical data, such as a corresponding block from a previous frame. Alternatively, "Inter" coding, in which the block of image data is encoded in terms of the differences between the block and a reference block of data, such as a corresponding block from a previous frame. Many variations on these two basic schemes are known to those skilled in the art, and thus are not discussed here in detail. It is generally desirable to select the encoding technique which requires the fewest number of bits to describe the block of data.

[0007] Intraframe encoding typically requires many more bits to represent the block. Therefore, interframe encoding is generally preferred. However there are some situations where the reference image block maintained at the receiver diverges from the corresponding reference block stored at the transmitter, such as when there are algorithmic differences in the implementation of the Inverse Discrete Cosine Transform (IDCT), or when transmission errors occur. Accordingly, when the transmitter encodes a block relative to a given reference, the block reconstructed by the receiver will differ from the block intended by the transmitter. It is therefore desirable that each block of data be coded in intraframe mode at least once for a given number of times that the block is coded in interframe mode. Details of one technique for such coding in the context of the H.261 standard are disclosed in U.S. Patent 5,644,660 to Bruder, which is hereby incorporated by reference in its entirety.

[0008] However, these prior art techniques are not suitable for application to newer coding standards, such as H.264. Particularly, in the H.264 video codec, unless the “constrained Intra” flag for the frame is set, Intra blocks are always predicted from the neighboring pixels. If the “constrained Intra” flag is set, all Intra blocks in the frame are only predicted from other Intra blocks, not necessarily from surrounding pixels. So, if one wants to gradually refresh the image by sending one or two Intra blocks each frame, one is given the undesirable choice of: (1) if the “constrained Intra” flag is clear, having image defect errors propagate into Intra regions due to the Intra prediction, or (2) if the “constrained Intra” flag is set, losing a significant benefit of the H.264 video codec by having all Intra blocks in the frame, whether they are refresh blocks or blocks that are more efficiently transmitted as Intra, constrained to only using neighboring Intra coded pixels.

[0010] Therefore, there is a need for a system and a method to provide improved Intra refresh while preserving the efficiency of the video codec, thereby improving video quality.

SUMMARY OF THE INVENTION

[0011] The present invention is directed to a method for a video encoder, by the use of classification maps, to transmit groups of pixels that are used to refresh discrepancies between an encoder's and decoder's reference frames. Because the groups of pixels are being used for what is essentially an error correction task, they cannot be based on information from other pixels, as opposed to groups of pixels that use image redundancies to improve coding efficiency. The H.264 standard articulates that only macroblocks within the same slice group may be spatially predicted off one another. H.264 also permits a map to be sent describing which slice group each macroblock in the frame is assigned to. By sending a map placing a small subset of macroblocks in one slice group and the remainder of the macroblocks in one or more other slice groups, one can produce the desired effect of isolating the refresh blocks of the picture from blocks that exploit image redundancies. Further, by sending a different map for each transmitted frame, each map corresponding with the macroblocks to be Intra refreshed in that frame, the effect of gradually refreshing all parts of the image can be achieved. Finally, by assigning a different frame index to each transmitted map, the map description only needs to be sent once at the start of the communication. All subsequent frames that use the same pattern of refresh blocks can reference the previously transmitted map index. The result is an efficiently transmitted self-correcting video sequence with only the additional channel overhead of sending the plurality of refresh maps at the start of the communication.

[0012] The invention maintains the highest level of video quality and compression rate while still giving the ability to clean up occasional line errors in H.264 conferences. Although the invention is described with reference to a video conferencing application, it is foreseen that the invention would also find beneficial application in other applications involving digitization of video data, e.g., the recording of DVDs, etc.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of an exemplary video conferencing system;

FIG. 2 is a block diagram of an exemplary video conference station of the video conferencing system of FIG. 1;

FIG. 3 is a block diagram of an exemplary embodiment of the image processing engine of FIG. 2.

FIG. 4 is a flow chart illustrating a method of encoding video data.

FIGS. 5A & 5B are block diagrams of video frames divided into a plurality of macroblocks and slices.

FIGS. 6A & 6B illustrate intra macroblock maps for the video frames of FIGS 5A and 5B.

DETAILED DESCRIPTION OF THE INVENTION

[0013] FIG. 1 illustrates an exemplary video conferencing system 100. The video conferencing system 100 includes a local video conference station 102 and a remote video conference station 104 connected through a network 106. Although FIG. 2 only shows two video conference stations 102 and 104, those skilled in the art will recognize that more video conference stations may be coupled to the video conferencing system 100. It should be noted that the present system and method may be utilized in any communication system where video data is transmitted over a network. The network 106 may be any type of electronic transmission medium, such as, but not limited to, POTS (Plain Old Telephone Service), cable, fiber optic, and radio transmission media.

[0014] FIG. 2 is a block diagram of an exemplary video conference station 200. For simplicity, the video conference station 200 will be described as the local video conference station 102 (FIG. 1), although the remote video conference station 104 (FIG. 1) may contain a similar configuration. In one embodiment, the video conference station 200 includes a display device 202, a CPU 204, a memory 206, at least one video capture device 208, an image processing engine 210, and a communication interface 212. Alternatively, other devices may be provided in the video conference station 200, or not all above named devices provided.

[0015] The at least one video capture device 208 may be implemented as a charge coupled device (CCD) camera, a complementary metal oxide semiconductor (CMOS) camera, or any other type of image capture device. The at least one video capture device 208 captures images of a user, conference room, or other scenes, and sends the images to the image processing engine 210. The image processing engine 210 will be discussed in more detail in connection with FIG. 3. Conversely, the image processing engine 210 also transforms received data packets from the remote video conference station 104 into a video signal for display on the display device 202.

[0016] FIG. 3 is an exemplary embodiment of the image processing engine 210 of FIG. 2. The image processing engine 210 includes a coding engine 302, a transport engine 304, configured to place each of the encoded macroblocks into a particular format for transmission across the network, and a communication buffer 306. In other embodiments of the invention, the transport engine may be a macroblock packetization engine or may be absent or may be

incorporated in the coding engine 302. Additionally, the image processing engine 210 may include more or fewer elements.

[0017] Initially, a video signal from the video capture device 208 (FIG. 2) enters the coding engine 302, which converts each frame (501, FIGS 5A & 5B) of video into a desired format, and transforms (step 401, FIG. 4) each frame of the video signal into a set of macroblocks (502, FIGS. 5A & 5B). A macroblock is a data unit that contains blocks of data comprising luminance and chrominance components associated with picture elements (also referred to as pixels). For example, in the H.264 standard, a picture is divided into slices. A slice is a sequence of macroblocks (or macroblock pairs if macroblock-adaptive frame/field decoding is in use). H.264 block sizes are different than H.261 and H.263, although the macroblock is still the same. For reference H.264 allows the macroblock to be broken up into different size components for Inter blocks, and even Intra blocks allow both a 16 pixel x 16 pixel mode and a 4 pixel x 4 pixel mode. The DCT/Quantization/IDCT is done on 4x4 blocks instead of 8x8 blocks as in H.261 and H.263. Each macroblock is comprised of one 16x16 luminance and two 8x8 chrominance sample arrays. A macroblock comprises four 8 x 8 blocks of luminance data and two corresponding 8 x 8 blocks of chrominance data in a 4:2:0 chroma sampling format. An 8 x 8 block of data is an eight-column by eight-row matrix of data, where each data corresponds to a pixel of the video frame.

[0018] However, it should be noted that the present invention is not limited to macroblocks as conventionally defined, but may be extended to any data unit comprising luminance and/or chrominance data. In addition, the scope of the present invention covers other sampling formats, such as a 4:2:2 chroma sampling format comprising four 8 x 8 blocks of luminance data and four corresponding 8 x 8 blocks of chrominance data, or a 4:4:4 chroma sampling format comprising four 8 x 8 blocks of luminance data and eight corresponding 8 x 8 blocks of chrominance data.

[0019] In addition, the coding engine 302 encodes each macroblock to reduce the number of bits used to represent the image content. Each macroblock may be “intra-coded” or “inter-coded,” and a video frame may be comprised of a combination of intra-coded and inter-coded macroblocks. Intra-coded macroblocks are encoded without use of information from other video frames, *i.e.*, intra-coded frames are coded only with reference to themselves. Alternatively, inter-coded macroblocks are encoded using temporal similarities (*i.e.*,

similarities that exist between a macroblock from one frame and a closely matched macroblock from a previously coded frame). The corresponding macroblock from a previous reference video frame need not be in an identical spatial position within the previous frame, but rather may comprise data associated with pixels that are spatially offset from the pixels associated with the given macroblock. This arises from the use of motion compensation techniques that are known to those skilled in the art, and thus the details are not reproduced here.

[0020] Coding engine 302 preferably intra-codes macroblocks of a frame using a refresh mechanism. The refresh mechanism is a deterministic mechanism to eliminate mismatches between the encoder and decoder reference frames by intra-coding a specific pattern of macroblocks for each frame. For future reference, a macroblock intra-coded via the refresh mechanism will be referred to as a refresh intra-coded macroblock. The details of a refresh mechanism are discussed in U.S. Patent Application serial no. 10/328,513, filed December 23, 2002, entitled “Dynamic Intra-coded Macroblock Refresh Interval for Video Error Concealment,” which is commonly owned with the present application and which is hereby incorporated by reference in its entirety.

[0021] Coding engine 302 preferably generates (step 404, FIG. 4) an intra-macroblock map (FIGS. 6A & 6B) that identifies which macroblocks in a coded video frame are intra-coded. After the intra-macroblock map is generated, the image processing engine 210 sends the map to the remote video conference station 104 (FIG. 1). The map may be sent as part of a picture header data associated with the coded video frame, for example, although other data fields may be used.

[0022] As noted above, each picture of a video sequence is divided into one or more slices. Each slice (503, FIGS. 5A & 5B) comprises some number of macroblocks (502, FIGS. 5A & 5B). The macroblock to slice group map (Fig. 6A & 6B) is a way of mapping macroblocks of a picture into slice groups. The macroblock to slice group map consists of a list of numbers, one for each coded macroblock, specifying the slice group to which each coded macroblock belongs. FIGS. 6A & 6B illustrate intra macroblock maps corresponding to the video frames illustrated in FIGS 5A & 5B in which a “1” illustrates a first slice group 503 to be intra refreshed and a 2 illustrates a second slice group (not shown, but comprising the remaining macroblocks) to be inter coded.

[0023] H.264 permits Flexible Macroblock Ordering, which is accomplished by specifying in the macroblock to slice group map what slice group each macroblock in the frame is assigned to. During the coding process, only macroblocks in the same slice group can be predicted off one another. By sending (step 402, FIG. 4) a plurality of maps (FIGS 6A & 6B), each map placing a different one or two macroblocks in one slice group and the remainder of the macroblocks in the frame in the other slice group (step 403, FIG. 4), and then indexing the appropriate map to correspond with the macroblocks to be Intra refreshed in the frame (step 404, FIG.4), the designer can produce the desired effect of refreshing parts of the picture without the risk of error propagation into the refreshed areas. Meanwhile coding efficiency is maintained in the remainder of the picture since all of the other macroblocks belong to the same slice group.

[0024] It is important to note that the intra-macroblock maps only need to be transmitted once during a video sequence/videoconference/movie. The H.264 standard requires the decoder to be capable of retaining up to 256 intra-macroblock maps simultaneously. After a map has been transmitted, the encoder simply needs to refer to that map by number for the decoder to recall which map is being used for that frame, thereby maintaining the highest level of coding efficiency.

[0025] The invention has been explained above with reference to exemplary embodiments. It will be evident to those skilled in the art that various modifications may be made thereto without departing from the broader spirit and scope of the invention. Further, although the invention has been described in the context of its implementation in particular environments and for particular applications, those skilled in the art will recognize that the present invention's usefulness is not limited thereto and that the invention can be beneficially utilized in any number of environments and implementations. The foregoing description and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.